Journal of Applied Communications

Volume 78 | Issue 1 Article 6

Introducing Today's SGML

Carol Sanders Reiner

Recommended Citation
Reiner, Carol Sanders (1994) "Introducing Today's SGML," *Journal of Applied Communications*: Vol. 78: Iss. 1. https://doi.org/10.4148/1051-0834.1397

Introducing Today's SGML

Abstract

A review of "Introducing Today's SGML" by Elizabeth Gilmore in *Technical Communication: Journal of the Society for Technical Communication*.

Book Reviews

"Introducing Today's SGML," by Elizabeth Gilmore in *Technical Communication: Journal of the Society for Technical Communication*, Vol. 40, No. 2, (Second Quarter, May 1993). Society for Technical Communication: 901 N. Stuart St., Suite 304, Arlington VA 22203, pp. 210-218.

Those of us who write, edit, or design publications may soon be facing the dilemma of preparing publications not only for print but for multiple output: CD-ROM, databases, and mixed media. But, how do we do this without reformatting for each output? Elizabeth Gilmore introduces SGML (Standard General Markup Language) as a possible solution and focuses on its relevance to publishing. Her article is the first of three in two special sections devoted to SGML.

She charges that creating good-looking documents on desktops is no longer enough. We want to reuse that information. And, if we think ahead, we must worry about what systems, programs, and computers, users on the other end are using.

What Is the Answer?

Integrating standard markup language into the publishing system may be the answer. Making documents interchangeable so that information creation, management, retrieval, printing, and display can be performed using the hardware and software best for each task is the rationale behind SGML. It is a standard for document description, published by the international Organization for Standardization and released in 1986.

In an SGML document, both markup (also called "tags") and content are represented in basic ASCII code. Since these 128 characters are read by most computers, SGML documents can be exchanged easily among most computers. SGML offers additional advantages:

1. It enables information to be used in multiple applications because the markup identifies information according to purpose rather than format.
2. It enables documents to be processed in many different ways formatted for print, saved to a database, displayed online, or combined with other media for mixed-media displays.
3. It is open-ended and can be adapted for hypertext or other new developments that come along.

4. It does not go out of date when publishing equipment or programs are updated if newer technologies support SGML.

Markup

Markup, at one time, referred to handwritten marks on a manuscript made by an editor or designer to tell the typesetter how the manuscript should be formatted. With computers, these instructions became imbedded in the text. So that they would not print, special control characters were used.

In the early 1980s, the Graphics Communications Association (GCA) created a standard markup language called "GenCode," which they hoped would be accepted by typesetters. The codes proposed were in plain, printable ASCI with names that reflected the typographical function in symbols that people could relate to (a paragraph set off by angle brackets, for example). The difficulty was defining a tag set broad enough for universal acceptance but not too unwieldy in size.

Concurrent with GCA's efforts, the American National Standards Institute (ANSI) developed a standard for coding documents based on a computer typesetting language Generalized Markup Language (GML) developed by Dr. Charles Goldfarb for IBM. Documents were represented as having a tree structure. It began at the top with a "book" and classified each module of information, called an element, into categories such as chapter, chapter title, paragraph, list, warning, etc. Editors and/or designers then specified how each element in the tree was to be typeset. This was called a "generic" markup.

The GCA and ANSI committees combined efforts to create a language that could be used to define usage rules for elements that could appear in classes of documents. The author uses a memo to illustrate the tree structure to give readers a grasp of how SGML works. It classifies elements in the document in an outline form to arrive at type sizes for various heads, etc., much the same way publication writers and editors organize information.

How Are Writers Affected?

Document structure, represented by a hierarchy of modular elements, is the SGML concept that affects writers. Writers primarily create and structure data. Writers use *content structures* words, sentences, paragraphs, lists, footnotes, references, headings, sections, chapters, glossaries, and appendices to organize facts and ideas to make them meaningful to their audiences. Writers and editors use an outline to structure the information. If writers grasp and use the concept of SGML, they will not have to be desktop

publishers as well. (Where does this leave editors and desktop publishers?) As more and more publishers move their data into SGML (there are now two), more and more software programs will appear on the market.

What Lies Ahead?

An increasing number of North American and European industries are looking to SGML as a strong first link of interrelated systems that store and reuse information with flexibility.

The U.S. Department of Defense has mandated that manuals for all existing weapons systems be converted to digital delivery and display although maintaining their page-oriented form. All new manuals are being written or delivered in a digital form completely independent of print-oriented formatting (SGML). Because many companies supply technology and accompanying documentation to this department, this will impact many corporations.

Airline, aerospace, automotive, financial, insurance, legal, manufacturing, and pharmaceutical corporations are adopting SGML to help them manage their information. Their publishing products are delivered in multiple ways (paper, online, CD-ROM), so they want to use data in multiple formats.

Gilmore sees the freeing of information for open exchange as critical in the '90s. She sees the day when SGML will be transparent and intuitive beneath our fingertips.

The second of the three articles, "Getting Your Data Into SGML" (pp. 219-225), weighs the costs and benefits of various conversion strategies from typesetting tapes to Optical Character Recognition and offers tips for creating a conversion plan. The third article, "SGML and *The New Yorker Magazine*" (pp. 226-229), is a case study. Five SGML developers created five different document type definitions (DTDs) for the *New Yorker* as an exercise presented at SGML '92.

As communication professionals in an information/computer age, we owe it to ourselves to become familiar with SGML as an emerging possible solution to the multiple output dilemma. Before we know it, both our clients and employers may be demanding output flexibility. Will we be ready?

Carol Sanders Reiner
University of Arkansas